WoT Thing Description
Protocol Binding Template for OCF

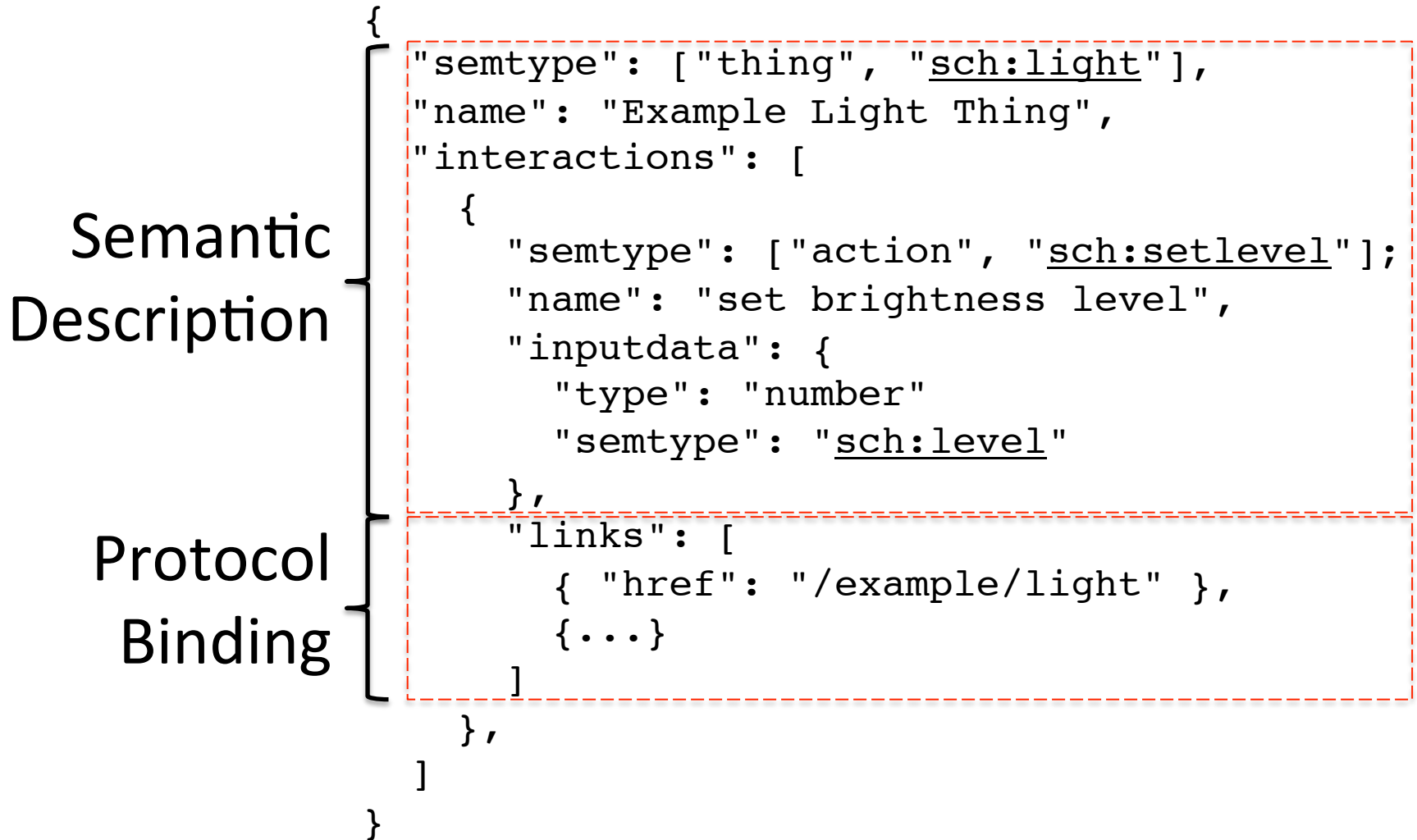Michael J Koster

May 5, 2017

Protocol Binding

- Maps abstract operations on WoT meta types to concrete operations on target resources
- WoT
  - Properties (get, set)
  - Actions (invoke, cancel)
  - Events (subscribe, unsubscribe)
- OCF
  - CRUD+N Resource Model, resource types, interface types, collections, actuation

Protocol Binding Template

- A specification for the information included in a particular protocol binding, and its format

- TD Interaction Description "body" contains information common to all protocol bindings

- Protocol Binding Template indicates which information is included per protocol and instance of the protocol binding to a resource (URI)

Thing Description (simplified)

Semantic Description

Protocol Binding

```
{
  "semtype": ["thing", "sch:light"],
  "name": "Example Light Thing",
  "interactions": [
    {
      "semtype": ["action", "sch:setlevel"];
      "name": "set brightness level",
      "inputdata": {
        "type": "number"
        "semtype": "sch:level"
    },
    "links": [
      { "href": "/example/light" },
      {...}
    ]
  },
 ]
}
```

OCF Protocol Binding

- Avoid embedding knowledge of specific OCF resource type constraints in software; provide enough information in the explicit controls to keep the "driver" generic – only need to generate mapping from RAML/Schemas to controls
  - oic.r.action proposed to identify high level actions
- Basic form of a hypermedia control, extension to the "links" property
  - link attributes describe high level operations
  - method, interface type, resource type, schema mapping to input/output data in TD

Generation of Thing Descriptions from OCF Introspection Data

- Generate TD and Protocol Bindings by mapping introspection data (Swagger) using template files and tools

- Dave Raggett has a tool that could create TD files from OCF resource descriptions

- OCF has an open source "swagger2x" tool that could create protocol bindings with input/output schemas

Strawman Example

- Highly explicit, identifies everything needed to use generic protocol engine that knows only CoAP, CBOR, and URI formatting
- Properties included in the OCF Protocol Binding
  - method (ocf.retrieve, ocf.update…)
  - ocf.rt - resource type
  - ocf.if - interface type
  - mediatype (default "application/vnd.ocf+cbor")
  - schemas for inputdata/outputdata elements

Alternate OCF Resource Formats

Composite Format

```
{
    "brightness": 30,
    "ramptime": 10
}
```

Batch Format

```
{
    "href":"",
    "rep": {
        "brightness": 30,
        "ramptime": 10
    }
}
```

TD – Interaction Description

```
{
  "semtype": ["action", "sch:setlevel"];
  "name": "set brightness level",
  "inputdata": {
    "type": "object"
    "properties": {
      "targetlevel": {
        "type": "number",
        "semtype": "sch:level"
      },
      "ramptime": {
        "type": "number",
        "semtype": "sch:transitiontime"
      }
    }
  },
  "links": [
    (protocol binding goes here)
  ]
}
```

JSON Schema + Extension

TD - Protocol Binding (Composite Format)

```
links: [
  {
    "href": "/example/light/brightness",
    "mediatype": "application/vnd.ocf+cbor",
    "method": "ocf.update",
    "ocf.rt": ["oic.r.brightness", "oic.r.ramptime"],
    "ocf.if": ["oic.if.a"],
    "inputschema": {
      "type": "object",
      "properties": {
        "brightness": {
          "type": "number",
          "value": "{{targetlevel}}"
        },
        "ramptime": {
          "type": "number",
          "value": "{{ramptime}}"
        }
      }
    }
  }
]
```

JSON Schema
+ Extension

(Batch Format)

```
"href": "/example/light/",
"method": "ocf.update",
"ocf.rt": ["oic.r.light"],
"ocf.if": ["oic.if.b"],
"inputschema": {
  "type": "object",
  "properties": {
    "href": {
      "type": "string",
      "value": ""
    },
    "rep": {
      "type": "object",
      "properties": {
        "brightness": {
          "type": "number",
          "value": "{{targetlevel}}"
        },
        "ramptime": {
          "type": "number",
          "value": "{{ramptime}}"
        }
      }
    }
  }
}
```

JSON Schema + Extension

Mapping of meta-operations

- Mapping to OCF CRUD+N model
- Interaction types have >1 meta-operation
- Property Get/Set maps to OCF Retrieve/ Update
- Action Invoke maps to OCF Update or Create
  - Action Cancel maps to OCF Delete if Created
- Event Subscribe maps to OCF Observe/Notify
  - Unsubscribe maps to GET without OBS option

Same Binding, Other Protocols

```
{
  "href": "mqtts://0m2m.net/example/light/actions",
  "method": "mqtt.publish",
  "mqtt.retain": true,
  "mqtt.qos": 1,
  "inputschema": {
    "type": "object",
    "properties": {
      "brightness": {
        "type": "number",
        "value": "{{targetlevel}}"
      },
      "ramptime": {
        "type": "number",
        "value": "{{ramptime}}"
      }
    }
  }
}
```

Examples

Examples in this presentation are in this TD file:

https://github.com/mjkoster/wot-protocol-binding/blob/master/td-ocf-protocol-binding.json

Other examples, alternate serializations, etc. may be found in:

https://github.com/mjkoster/wot-protocol-binding